Amendments to the Claims:

This listing of claims will replace all prior versions, and listing of claims in the application:

Listing of the Claims:

1. **(Currently Amended)** In a Java™ computing environment, an internal class representation suitable for use by a Java™ virtual machine, said internal class representation comprising:

a reference identifier having one or more entries, wherein each of said one or more entries correspond to a field of a Java™ object; and

wherein each of said one or more entries can be used to indicate whether **a** corresponding field**[[s]]** of said Java™ object is a reference to another Java™ object.

2. (Original) An internal class representation as recited in claim 1,

wherein said reference identifier is an array of bytes; and

wherein the size of said reference identifier is the same as the number of fields of said Java™ object.

3. **(Currently Amended)** An internal class representation as recited in claim **[[1]] 2**, wherein an entry of said array of bytes is set to zero to indicate that the corresponding field of said Java™ object is not a reference to another Java™ object.

A1

4. **(Currently Amended)** ~~An internal class representation as recited in claim 1~~, **In a Java™ computing environment, an internal class representation suitable for use by a Java™ virtual machine, said internal class representation comprising:**

a reference identifier having one or more entries, wherein each of said one or more entries correspond to a field of a Java™ object; and

wherein each of said one or more entries can be used to indicate whether a corresponding field of said Java™ object is a reference to another Java™ object;

wherein said reference identifier is an array of bytes,

wherein the size of said array of bytes is the same as the number of fields of said Java™ object, and

wherein an entry of said array of bytes is set to a predetermined non-zero value to indicate that the corresponding field of said Java™ object is not a reference to another Java™ object.

5. (Original) An internal class representation as recited in claim 4, wherein said predetermined non-zero value is equal to 1.

6. **(Currently Amended)** An internal class representation as recited in claim **[[1]] 2**,

wherein an entry of said array of bytes is set to zero to indicate that the corresponding field of said Java™ object is not a reference to another Java™ object; and

wherein an entry of said array of bytes is set to a predetermined non-zero value to indicate that the corresponding field of said Java™ object is not a reference to another Java™ object.

7. **(Currently Amended)** An internal class representation as recited in claim [[6]] **4**, wherein said array of bytes is allocated and set to appropriate values during load time.

8. (Original) An internal class representation as recited in claim 1, wherein said reference identifier is allocated during load time.

A1

9. **(Currently Amended)** A method for generating a reference identifier for a Java™ object, said method comprising:

reading a class file associated with a Java™ object;

identifying fields of said Java™ object that are references;

allocating a reference identifier for said Java™ object; and

~~wherein~~ **initializing** said reference identifier **to indicate** ~~indicates~~ which fields of said Java™ object are references.

10. (Original) A method as recited in claim 9, wherein said method is performed at load time by a virtual machine.

11. (Original) A method as recited in claim 9,

wherein said reference identifier is an array of bytes; and

wherein the size of said reference identifier is the same as the number of fields of said Java™ object.

12. (Original) A method as recited in claim 9, wherein an entry of said array of bytes is set to zero to indicate that the corresponding field of said Java™ object is not a reference to another Java™ object.

13. (Original) A method as recited in claim 9, wherein an entry of said array of bytes is set to a predetermined non-zero value to indicate that the corresponding field of said Java™ object is not a reference to another Java™ object.

14. (Currently Amended) A method for determining whether a field of a Java™ object is a reference to another Java™ object, said method comprising:

identifying [[the]] **an** internal class representation for [[the]] **a** Java™ object;

identifying a reference ~~indicator~~ **identifier** in said internal class representation;

reading a portion of said reference ~~indicator~~ **identifier** that represents said field of said Java™ object; and

determining whether the value stored in said portion of said reference ~~indicator~~ **identifier** is equal to a predetermined value.

A1

15. (Original) A method as recited in claim 14, wherein said method is performed by a Java™ virtual machine at runtime.

16. (Original) A method as recited in claim 14,

wherein said reference identifier is an array of bytes; and

wherein the size of said reference identifier is the same as the number of fields of said Java™ object.

17. (Original) A method as recited in claim 14, wherein said predetermined value can be 1 or zero.

18. (Original) A computer readable media including computer program code for an internal class representation suitable for use by a Java™ virtual machine, said computer readable media comprising:

computer program code for a reference identifier having one or more entries, wherein each of said one or more entries correspond to a field of a Java™ object; and

wherein each of said one or more entries can be used to indicate whether corresponding fields of said Java™ object is a reference to another Java™ object.

19. (Original) A computer readable media as recited in claim 18,

wherein said reference identifier is an array of bytes; and

wherein the size of said reference identifier is the same as the number of fields of said Java™ object.

A1

20. (Original) A computer readable media as recited in claim 19, wherein said array of bytes is allocated and set to appropriate values during load time.